

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-113887

(43)Date of publication of application : 07.05.1993

(51)Int.Cl.

G06F 9/46

G06F 9/46

(21)Application number : 03-274285

(71)Applicant : NEC CORP

(22)Date of filing : 22.10.1991

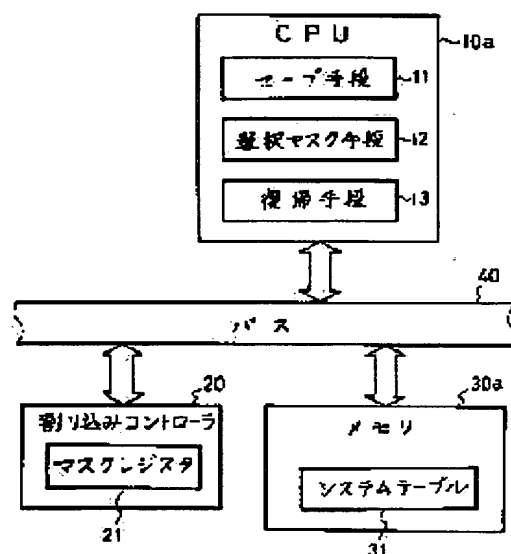
(72)Inventor : OKUDA RIKI

(54) COMPUTER SYSTEM

(57)Abstract:

PURPOSE: To accelerate interruptive responsiveness by shortening interruption prohibiting time in the system call processing of a real time operating system.

CONSTITUTION: Designated mask data in which only a bit of arbitrary interruption level is set is registered on a system table 31, and the present interruption level value of a mask register 21 is saved in the system table 31 by a saving means 11, and the AND of the saved value of the mask register 21 and a designated mask data value is taken by a selection mask means 12, and a new designated mask data value is set at the value of the mask register 21, then, the arbitrary interruption level is masked. After that, the value of the mask register 21 is returned to a certain value by saving after making access the resources by a restoring means 13.



LEGAL STATUS

[Date of request for examination] 27.10.1995

[Date of sending the examiner's decision of rejection] 09.12.1997

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平5-113887

(43)公開日 平成5年(1993)5月7日

(51)Int.Cl.⁵

G 0 6 F 9/46

識別記号

3 1 1 E

庁内整理番号

8120-5B

3 4 0 F

8120-5B

F I

技術表示箇所

審査請求 未請求 請求項の数1(全 9 頁)

(21)出願番号 特願平3-274285

(22)出願日 平成3年(1991)10月22日

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 奥田 利花

東京都港区芝五丁目7番1号 日本電気株式会社内

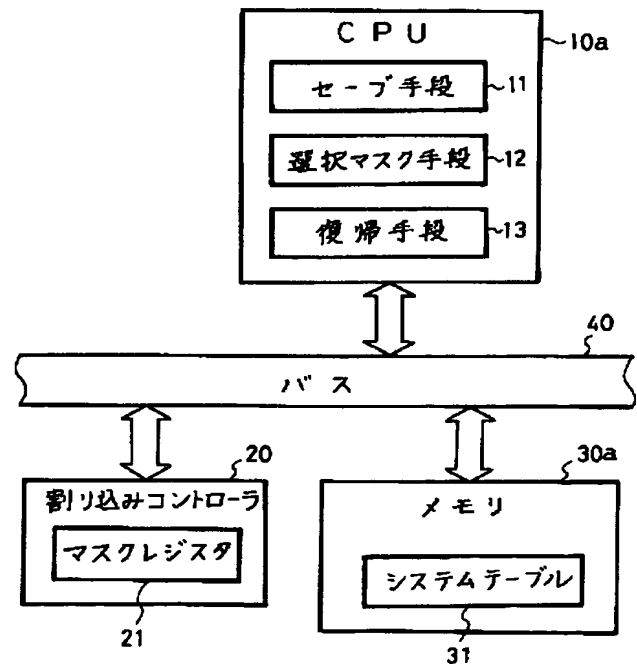
(74)代理人 弁理士 井出 直孝

(54)【発明の名称】 コンピュータシステム

(57)【要約】

【目的】 リアルタイムオペレーティングシステムのシステムコール処理中の割り込み禁止時間を短縮し割り込み応答性の向上を図る。

【構成】 任意の割り込みレベルのビットのみを立てた指定のマスクデータをシステムテーブル31に登録しておき、セーブ手段11によりマスクレジスタ21の現在の割り込みレベル値をシステムテーブル31にセーブし、選択マスク手段12によりこのセーブされたマスクレジスタ21の値と前記指定のマスクデータ値との論理和をとり、前記指定のマスクデータ値を新しくマスクレジスタ21の値に設定し前記任意の割り込みレベルをマスクする。そして、復帰手段13により資源アクセス終了後にマスクレジスタ21の値をセーブしてある元の値に戻す。



【特許請求の範囲】

【請求項1】 割り込みレベルを記憶するマスクレジスタを有する割り込みコントローラと、メモリと、中央処理装置とを備えたコンピュータシステムにおいて、前記メモリは、任意の割り込みレベルのビットを立てたマスクデータが登録されたシステムテーブルを含み、前記中央処理装置は、前記マスクレジスタの割り込みレベルの現在の値を前記システムテーブル中にセーブするセーブ手段と、このセーブされた割り込みレベル値と前記システムテーブルに登録されたマスクデータ値との論理和をとり前記マスクデータ値を前記マスクレジスタの新しい割り込みレベル値として設定し前記任意の割り込みレベルをマスクする選択マスク手段と、資源アクセス終了後に前記マスクレジスタの値をセーブしてある元の値に戻す復帰手段とを含むことを特徴とするコンピュータシステム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、リアルタイムに動作するコンピュータシステム（以下、オペレーティングシステムという。）に利用され、特に、割り込みの迅速な応答性を要求されるリアルタイムオペレーティングシステムの資源アクセス中の割り込み制御方式に関する。

【0002】

【従来の技術】従来の、この種の資源アクセス中の割り込み制御処理方式を、タスクを強制終了するシステムコール終了処理タスク〔外1〕を例にとり、図面を参照しながら説明する。

【0003】

【外1】

TER_TSK

図6は従来例の要部を示すブロック構成図である。図6において、10はCPU（中央処理装置）、20は割り込みレベルを記憶するマスクレジスタ21を有する割り込みコントローラ、30はメモリ、および40はバスである。

【0004】次に、この従来例の動作について図7に示すフローチャートを参照して説明する。

【0005】リアルタイムオペレーティングシステムのタスク、割り込みハンドラのコードとデータはメモリ30上にある。システム上にAおよびBの二つのタスクが登録されている。これらのタスクは、レディ（READY、動作可能）状態、待ち（WAIT）状態、中断（SUSPEND）状態、および休止（DORMANT）状態の四つの状態を取り得るが、現在の状態はレディ状態でプライオリティはタスクBの方が高い。タスクBはタスクAを強制終了するシステムコール終了処理タスク

（図7のフローチャート）を発行する。割り込みコントローラ20のレベル1および2には割り込みハンドラが割り付けられている。レベル1に対応する割り込みハン

ドラCはシステムコールは発行せず、レベル2に対応する割り込みハンドラDはタスクAの状態を中断状態にするシステムコール中断処理タスク〔外2〕を発行する。CPU10は割り込み受け付け可能（EI）状態で、割り込みコントローラ20のレベル1および2は割り込みマスクがされていないで、プライオリティはレベル1の方が高いと仮定する。

【0006】

【外2】

SUS_TSK

この状態で、プライオリティの高いタスクBが動き出し、タスクAを強制終了するシステムコール終了処理タスクを発行する。図7のステップS21でCPU10に対し、全てのマスカブル割り込みの受け付けを禁止するDI命令を発行する。次に、ステップS22でタスクAの状態を調べて分岐する。ここでレベル1および2の割り込みが発生したと仮定すると、全てのマスカブル割り込みは禁止されているため二つの割り込みとも受け付けられない。従って、タスクAを強制終了させる処理が続けられ、ステップS23でタスクAのコントロールテーブルをレディキューから外し、ステップS25でタスクAの状態を休止状態とする。そしてステップS26でCPU10に対しEIを発行して、全てのマスカブル割り込みを許可する。ここで、レベル1の割り込みが受け付けられ、割り込みハンドラCが処理される。ハンドラCの処理が終わると、レベル2の割り込みが受け付けられハンドラDの処理に移り、システムコール中断処理は「指定タスクの状態が休止状態である」というエラーを返して終了する。割り込みハンドラDの処理が終了すると、タスクAを強制終了させる処理の続きが再開し、ステップS27でコントロールテーブルの内容を初期化する。

【0007】このように資源をアクセスする間（ステップS22の分岐処理が終わってからステップS26の処理が終わるまで）、全てのマスカブル割り込みを禁止することにより、同一資源へのアクセスの重複を防いでいる。

【0008】

【発明が解決しようとする課題】前述した従来のリアルタイムオペレーティングシステムにおける資源アクセス中の割り込み制御方式は、ある資源に対するシステムコール処理を発行する割り込みハンドラのみを禁止することを目的に、割り込みハンドラ全てを禁止している。ここで割り込みを禁止する必要があるのは割り込みハンドラの中でシステムコールを発行するものだけである。システムコールを発行しない割り込みレベルは割り込みを禁止する必要がない。リアルタイムオペレーティングシステムを使用するような分野では、割り込みの応答性（割り込み信号が発生してからCPUに受け付けられるまでの時間）が最も重要な性能の一つである。にもかか

10

20

30

40

50

ならず、前述した従来の割り込み禁止処理方式はある種の割り込みを禁止するために、その他全ての割り込みを不必要に禁止しており、割り込みの応答性を著しく下げる欠点があった。

【0009】本発明の目的は、前記の欠点を除去することにより、システムコール中の割り込み禁止時間を短縮し割り込みの応答性を向上させたリアルタイムオペレーティングシステムを提供することにある。

【0010】

【課題を解決するための手段】本発明は、割り込みレベルを記憶するマスクレジスタを有する割り込みコントローラと、メモリと、中央処理装置とを備えたコンピュータシステムにおいて、前記メモリは、任意の割り込みレベルのビットを立てたマスクデータが登録されたシステムテーブルを含み、前記中央処理装置は、前記マスクレジスタの割り込みレベルの現在の値を前記システムテーブル中にセーブするセーブ手段と、このセーブされた割り込みレベル値と前記システムテーブルに登録されたマスクデータ値との論理和をとり前記マスクデータ値を前記マスクレジスタの新しい割り込みレベル値として設定し前記任意の割り込みレベルをマスクする選択マスク手段と、資源アクセス終了後に前記マスクレジスタの値をセーブしてある元の値に戻す復帰手段とを含むことを特徴とする。

【0011】

【作用】本発明は、各手段により、システムコールを発行する割り込みハンドラのレベルのビットのみ立てたマスクデータを持ち、従来、全ての割り込みを禁止していた資源アクセスの箇所で、前記のマスクデータを用いシステムコールを発行する割り込みレベルのみマスクするように制御を行う。

【0012】従って、従来全ての割り込みを禁止していたのに比べ割り込み禁止時間を大幅に短縮することができ、割り込みの応答性を向上させることが可能となる。

【0013】

【実施例】以下、本発明の実施例について図面を参照して説明する。

【0014】図1は本発明の一実施例を示すブロック構成図、および図2はそのシステムテーブルの一例を示す説明図である。

【0015】本実施例は、CPU10aと、割り込みレベルを記憶するマスクレジスタ21を有する割り込みコントローラ20と、メモリ30aと、それらを接続するバス40とを備えたリアルタイムオペレーティングシステムにおいて、本発明の特徴とするところの、メモリ30aは、任意の割り込みレベルのビットを立てたマスクデータ31aとマスクレジスタ値セーブ領域31bとを含むシステムテーブル31を有し、CPU10aは、マスクレジスタ21の現在の値をシステムテーブル31のマスクレジスタ値セーブ領域31bにセーブするセーブ

手段11と、このセーブされたマスクレジスタ21の現在の値とシステムテーブル31に登録されたマスクデータ31aの値との論理和をとりマスクデータ31aの値をマスクレジスタ21の新しい割り込みレベル値として設定し前記任意の割り込みレベルをマスクする選択マスク手段12と、資源アクセス終了後にマスクレジスタ21の値をセーブしてある元の値に戻す復帰手段13とを含んでいる。なお、図2において31bはマスクレジスタ値セーブ領域である。

【0016】次に、本実施例の動作例(1)について図3に示すフローチャートを参照して説明する。

【0017】リアルタイムオペレーティングシステムのシステムが立ち上がり、A、B二つのタスクが登録されており、それぞれレディ状態でプライオリティはタスクBの方が高いとする。タスクBはタスクAに対してシステムコール終了処理タスク(図3のフローチャート)を発行する。割り込みコントローラ20のレベル1およびレベル2には割り込みハンドラが割り付けられている。レベル1に対応する割り込みハンドラCはシステムコールは発行せず、レベル2に対応する割り込みハンドラDはタスクAに対して中断処理タスクを発行する。CPU10aはEI状態で、割り込みコントローラ20のレベル1およびレベル2は割り込みマスクがされていないで、プライオリティはレベル1の方が高い。図3システムテーブル31にあるマスクデータ31aは、システムコールを発行するレベル2の割り込みだけをマスクする値になっていると仮定する。

【0018】プライオリティの高いタスクBが動き出し、タスクAに対してシステムコール終了処理タスクを発行する。図3のステップS1で、割り込みコントローラ20の現在のマスクレジスタ21の値をシステムテーブル31のマスクレジスタ値セーブ領域31bにセーブする。次に、ステップS2で現在のマスクレジスタ21の値とマスクデータ31aの値の論理和をマスクレジスタ21の新しい値とする。

【0019】ここで、レベル1およびレベル2の割り込みが発生したと仮定すると、レベル1はマスクされていないので割り込みが受け付けられ、処理を行う。ハンドラCの処理が終了しても、レベル2の割り込みはマスクされているので、受け付けられない。タスクAを強制終了させる処理の続きが再開し、ステップS3でタスクAの状態はレディ状態なのでステップS4へ分岐し、ステップS4でタスクAのコントロールテーブルをレディキューから外し、ステップS6で状態を休止状態とする。そして、ステップS7でマスクレジスタ21の値をセーブしておいた元の値に戻す。ここで、レベル2の割り込みが受け付けられ、割り込みハンドラCはタスクAを中断しようとするが、タスクAの状態は休止状態なので、対象タスクが休止状態だったというエラーコードを持って終了する。割り込み処理が終わると、タスクAを強制

終了させる処理の続きが再開し、ステップS 8でコントロールテーブルの初期化を行う。

【0020】以上のようにして、システムコール処理および割り込み処理が行われる。

【0021】次に、本実施例の動作例(2)について図4および図5を参照して説明する。

【0022】この動作例(2)においては、システムテーブルとして図4に示すシステムテーブル32を使用する。動作例(2)ではマスクデータが資源の種類別に分かれており、図4に示すように、システムテーブル32は、フラグをアクセスする割り込みのマスクデータ32a、セマフォをアクセスする割り込みのマスクデータ32b、メールボックスをアクセスする割り込みのマスクデータ32c、タスクをアクセスする割り込みのマスクデータ32d、タイマーをアクセスする割り込みのマスクデータ32e、およびマスクレジスタ値セーブ領域32fを含んでいる。

【0023】リアルタイムオペレーティングシステムのシステムが立ち上がり、A、B二つのタスクが登録されており、それぞれレディ状態でプライオリティはタスクBの方が高いとする。タスクBはシステムコール終了処理タスク(図5のフローチャート)を発行する。割り込みコントローラ20のレベル1、レベル2およびレベル3には割り込みハンドラが割り付けられている。レベル1に対応する割り込みハンドラCはシステムコールを発行しない。レベル2に対応する割り込みハンドラDはシステムコール中断処理タスクを発行する。レベル3に対応する割り込みハンドラEはイベントフラグをセットするシステムコールセットフラグ(〔外3〕)を発行する。CPUはE I状態で、割り込みコントローラ20のレベル1、レベル2およびレベル3は割り込みマスクがされていないで、プライオリティはレベル1、レベル2、レベル3の順に高い。フラグをアクセスする割り込みレベルのマスクデータ32a(図4)の値はレベル3のビットを立てた値で、タスクをアクセスする割り込みレベルのマスクデータ32d(図4)はレベル2のビットを立てた値になっていると仮定する。

【0024】

〔外3〕

SET_FLG

プライオリティの高いタスクBが動き、タスクAに対してシステムコール終了処理タスクを発行する。図5のステップS 11で現在のマスクレジスタ21の値をシステムテーブル32中のマスクレジスタ値セーブ領域32fにセーブする。次に、ステップS 12でタスクをアクセスする割り込みのマスクデータ32dを選択する。そして、ステップS 13で現在のマスクレジスタ21の値とマスクデータ32dの値との論理和をとり、その結果を新しくマスクデータ値とする。

【0025】ここでレベル1、レベル2およびレベル3

の割り込みが発生したと仮定すると、プライオリティの一番高いレベル1はマスクされていないので割り込みが受け付けられ、処理を終える。2番目のプライオリティの高いレベル2の割り込みはマスクされているため割り込みが受け付けられない。レベル3はマスクされていないので割り込みが受け付けられ、システムコールセットフラグを発行し、処理を終える。割り込み処理が終わると、タスクAを強制終了させる処理の続きが再開し、ステップS 14でタスクAの状態がレディ状態なのでステップS 15へ飛び、ステップS 15でタスクAのコントロールテーブルをレディキューから外し、ステップS 17でタスクAの状態を休止状態にする。そして、ステップS 18でマスクレジスタ21の値をセーブしておいた元の値に戻す。

【0026】ここで、ステップS 16でレベル2の割り込みが遅延して受け付けられ、割り込みハンドラCはタスクAを中断しようとするが、タスクAの状態は休止状態なので、対象タスクが休止状態だったというエラーコードを持って終了する。割り込み処理が終わると、タスクAを強制終了させる処理の続きが再開し、ステップS 19でタスクのコントロールテーブルを初期化する。

【0027】

【発明の効果】以上説明したように本発明は、システムコール処理で資源をアクセスする際に、システムコールを発行する割り込みレベルのみを前述のマスクデータを用いてマスクすることにより、システムコールを発行しない割り込みレベルに対して、割り込みの応答遅延時間が0になり、リアルタイムオペレーティングシステムにとって一番大切な性能である割り込みに対する応答性が非常によくなる効果がある。

【0028】また、ハンドラ中で発行するシステムコールのアクセスする資源別にマスクデータを細かく分ければ、システムコールを発行する割り込みレベルの中でも、その資源をアクセスするレベルのみをマスクするので、その他のレベルを不必要にマスクせず、割り込みの応答性がさらによくなる効果が得られる。

【図面の簡単な説明】

【図1】本発明の一実施例の要部を示すブロック構成図。

【図2】そのシステムテーブルの一例を示す説明図。

【図3】その動作例(1)を示すフローチャート。

【図4】そのシステムテーブルの他の例を示す説明図。

【図5】その動作例(2)を示すフローチャート。

【図6】従来例を示すブロック構成図。

【図7】その動作を示すフローチャート。

【符号の説明】

10、10a CPU

11 セーブ手段

12 選択マスク手段

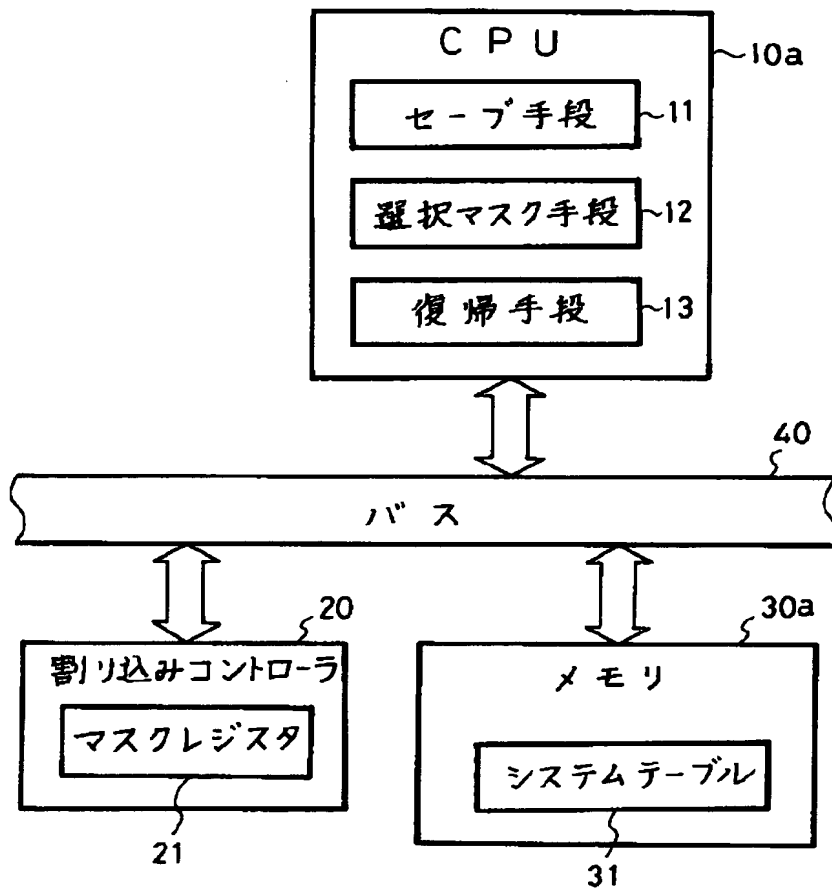
13 復帰手段

7
 20 割り込みコントローラ
 21 マスクレジスタ
 30、30a メモリ
 31、32 システムテーブル
 31a、32a～32e マスクデータ
 31b、32f マスクレジスタ値セーブ領域

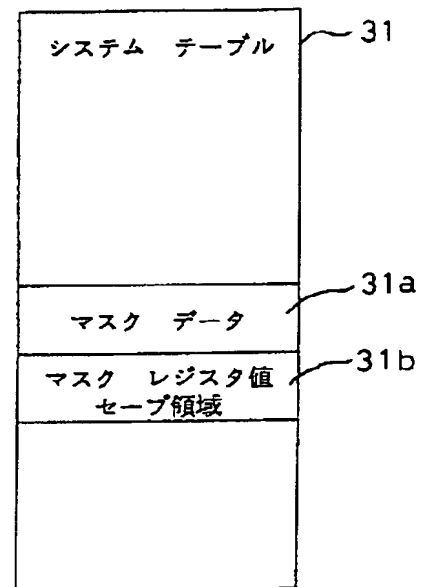
8
 * 31b、32f マスクレジスタ値セーブ領域
 40 バス
 S1～S8、S11～S19、S21～S27 ステップ

*

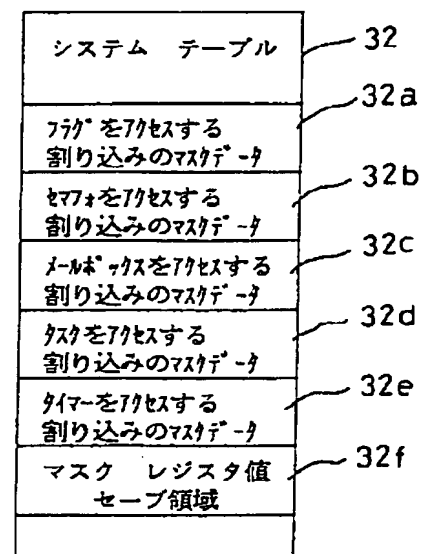
【図1】



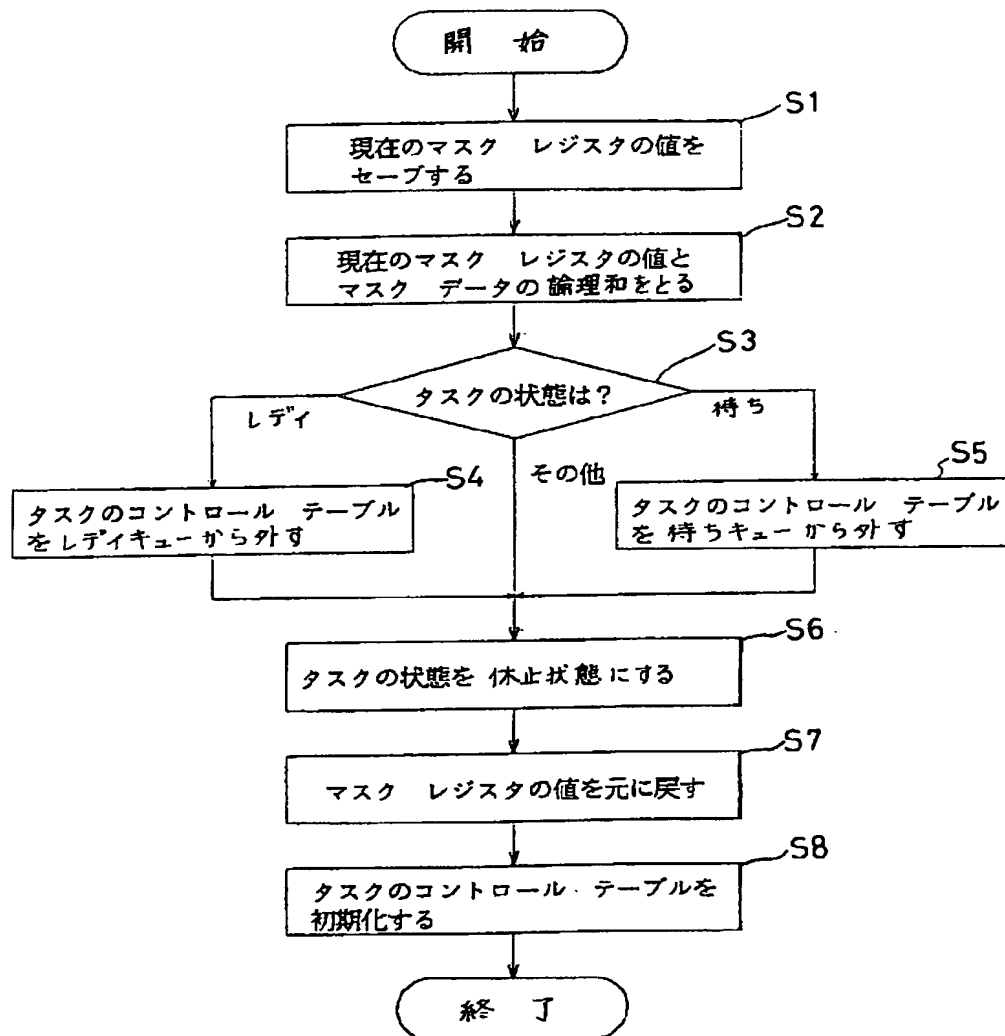
【図2】



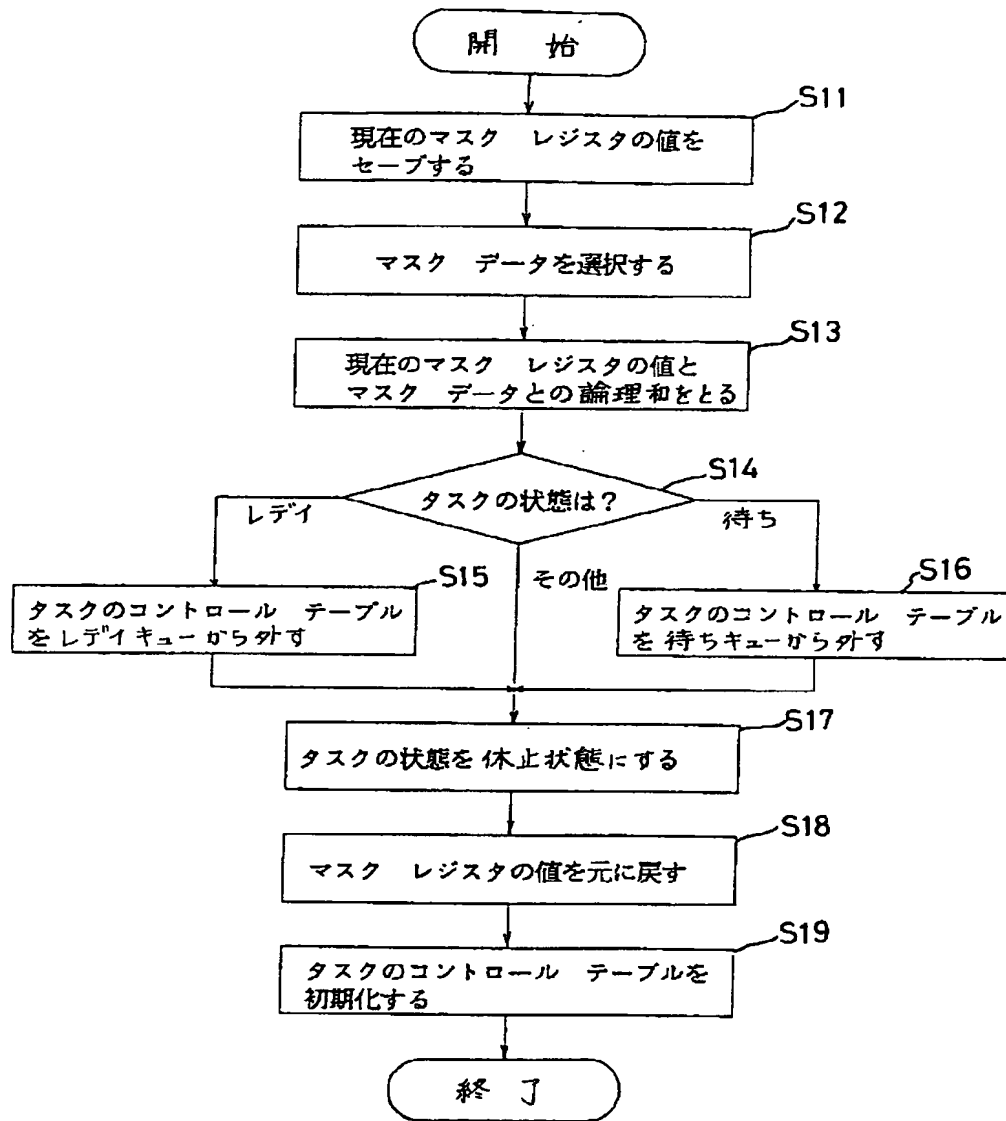
【図4】



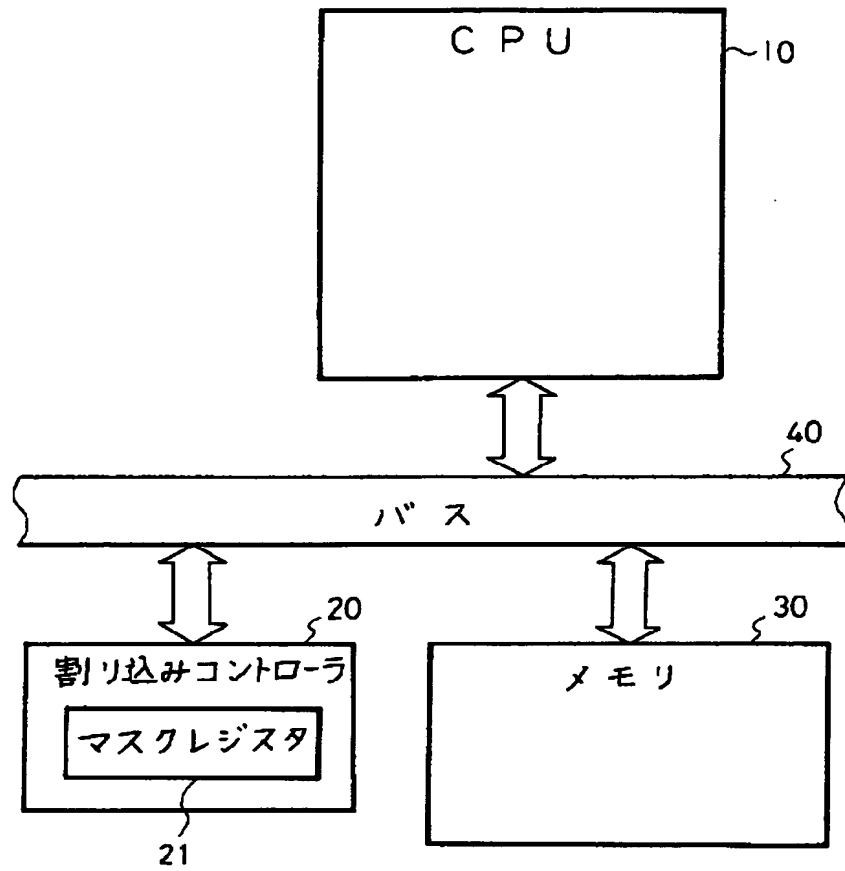
【図3】



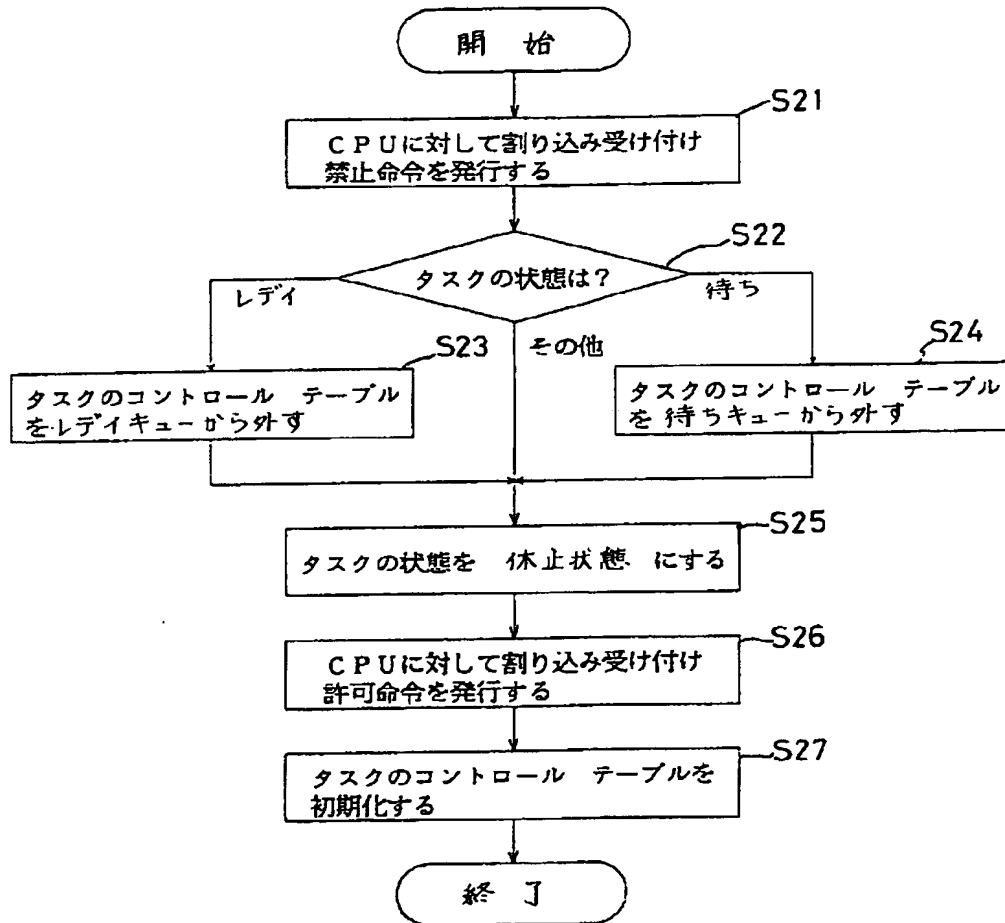
【図5】



【図6】



【図7】

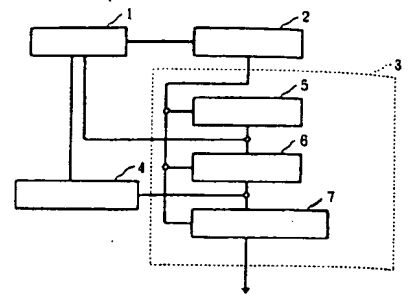


(54) PARALLEL PROCESSING PROGRAM DEVELOPING DEVICE

(11) 5-113885 (A) (43) 7.5.1993 (19) JP
 (21) Appl. No. 3-275283 (22) 23.10.1991
 (71) NITTO SEIKO CO LTD (72) YOSHIHIRO HAYASHI
 (51) Int. Cl⁵. G06F9/45

PURPOSE: To obtain a parallel processing program developing device capable of generating a program efficiently by extracting parallelism sufficiently.

CONSTITUTION: A parallel processing program by language including a parallel execution text in which processing being executed simultaneously are unified is inputted to an input device 2. An analysis means 5 analyzes the parallel processing program inputted via the input device 2, and judges the required number of arithmetic means and processing with which each arithmetic means should charge. A conversion means 6 converts the parallel processing program to a program to operate each arithmetic means by receiving the decided number of arithmetic means and the processing with which each arithmetic means should charge via the input device 2 by referring to the judged result of the analysis means 5. A code generating means 7 converts the program converted by the conversion means 6 to an instruction code.



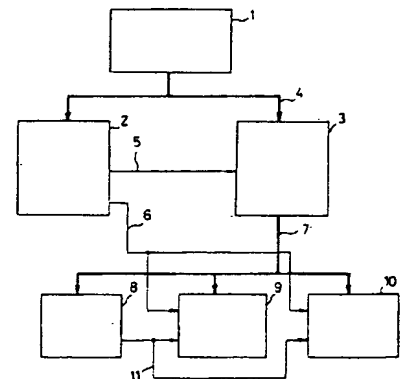
1: display device, 3: compiler, 4: simulator

(54) COMPUTER

(11) 5-113886 (A) (43) 7.5.1993 (19) JP
 (21) Appl. No. 3-275771 (22) 23.10.1991
 (71) NEC CORP (72) TETSUYA FUJITA
 (51) Int. Cl⁵. G06F9/45, G06F9/38

PURPOSE: To provide a computer capable of executing a parallel instruction as maintaining interchangeability with an existing computer without increasing hardware quantity in a decoder part.

CONSTITUTION: This computer is equipped with a parallel instruction analysis compiler which stores control information with respect to parallel instruction execution control in an object code by utilizing an undefined instruction word, a parallel information decoder unit 2 which decodes the control information to be executed in parallel by taking out from the undefined instruction word, and an instruction decode issuing unit 3 which issues a single or plural instructions to a branch instruction execution unit 8 based on the control information.



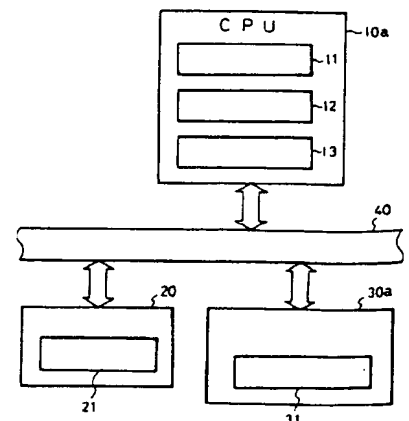
1: instruction buffer, 3: instruction decode issuing unit,
 4,7: instruction transmission means, 5: parallel instruction
 issuing signal means, 6: instruction cancellation signal means,
 9: integer arithmetic unit, 10: floating point arithmetic
 unit, 11: conditional branch signal means

(54) COMPUTER SYSTEM

(11) 5-113887 (A) (43) 7.5.1993 (19) JP
 (21) Appl. No. 3-274285 (22) 22.10.1991
 (71) NEC CORP (72) RIKI OKUDA
 (51) Int. Cl⁵. G06F9/46

PURPOSE: To accelerate interruptive responsiveness by shortening interruption prohibiting time in the system call processing of a real time operating system.

CONSTITUTION: Designated mask data in which only a bit of arbitrary interruption level is set is registered on a system table 31, and the present interruption level value of a mask register 21 is saved in the system table 31 by a saving means 11, and the AND of the saved value of the mask register 21 and a designated mask data value is taken by a selection mask means 12, and a new designated mask data value is set at the value of the mask register 21, then, the arbitrary interruption level is masked. After that, the value of the mask register 21 is returned to a certain value by saving after making access the resources by a restoring means 13.



20: interruption controller, 40: bus, 30a: memory